

TERAKREDITASI RISTEKDIKTI No. 36b/E/KPT/2016

Jurnal *Rekayasa Elektrika*

VOLUME 14 NOMOR 3

Desember 2018

Pengendalian Kecepatan Gerak Quadcopter Berbasis IMU pada Gerakan Tangan dengan Metode Template Matching 205-213

Nofria Hanafi, Abdul Mu'iz Miftahudin, Naufal Ali Akbar C, dan R. Sanggar Dewanto

JRE	Vol. 14	No. 3	Hal 145–220	Banda Aceh, Desember 2018	ISSN. 1412-4785 e-ISSN. 2252-620X
-----	---------	-------	-------------	------------------------------	--------------------------------------

Pengendalian Kecepatan Gerak Quadcopter Berbasis IMU pada Gerakan Tangan dengan Metode Template Matching

Nofria Hanafi, Abdul Mu'iz Miftahudin, Naufal Ali Akbar C, dan R. Sanggar Dewanto
Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jl. Raya ITS Sukolilo, Surabaya 60111
e-mail: hanafi@pens.ac.id

Abstrak—Wahana udara berpenggerak 4 motor dc *brushless (quadcopter)* dioperasikan menggunakan *remote control* atau otomatis. Wahana udara dengan sebutan *quadcopter* dikendalikan dengan memanfaatkan data sensor *Inertial Measurement Unit (IMU)* terutama sensor akselerometer dan giroskop. Pengendali gerakan berupa tongkat kendali dengan data sensor IMU sebagai sumber data masukan. Pergerakan wahana udara merupakan pergerakan dua dimensi (2D) yaitu pada sumbu X dan Y. Data sensor IMU dibaca oleh mikrokontroler dan diproses untuk mendapatkan data akselerasi (akselerasi tanpa terpengaruh gravitasi) kemudian dilakukan pengintegralan ganda terhadap data real akselerasi untuk mendapatkan trayektori gerakan yang dilakukan oleh remote. Data hasil proses mikrokontroler dikirimkan ke PC (Personal Komputer) untuk proses rekognisi *gesture* dengan menggunakan metode *template matching*. Jarak pergerakan tongkat pengendali (*remote gesture inertia*) dan kecepatan mengoperasikannya merupakan parameter yang dikonversi menjadi gerakan wahana udara. Gerakan trayektori diubah dalam domain gambar berukuran 10x10 *pixel* sebagai pembandingan dengan algoritma *template matching*. Hasil gerakan rekognisi yang dikenali menjadi perintah *command* pergerakan ke *quadcopter*. Gerakan sederhana lurus vertikal, horizontal, diagonal, dan melengkung berhasil dilakukan dengan tingkat keberhasilan 96% menggunakan estimasi trayektori dan algoritma *template matching*.

Kata kunci: *Inertial Measurement Unit, Algoritma Template Matching, Quadcopter, Gesture inertia*

Abstract—An aerial vehicle has actuator four brushless DC motors (*quadcopter*) which is operated using a remote control or automatically. An aerial vehicle commonly called a *quadcopter* is controlled by using sensor data *Inertia Measurement Unit (IMU)* especially gyroscope and accelerometer sensor. Motion control is a stick control with sensor data IMU as data input. The motion from the aerial vehicle is a two-dimension (2D) motion from X axis and Y axis. Data sensor IMU is read by a microcontroller and will be processed to get real data acceleration (acceleration without influenced gravitation) remotely. The distance gesture inertia and the speed at which it operates are the parameter that is converted into an air vehicle motion. Trajectory motion is changed into picture domain 10x10 *pixel* as the comparison with the template matching algorithm. Match motion recognition is a motion command to the *quadcopter*. The simple motion like vertical, horizontal, diagonal, and curve match 96% using template matching algorithm and trajectory estimation.

Keywords: *Inertial Measurement Unit, Template Matching Algorithm, Quadcopter, Gesture inertia*

Copyright © 2018 Jurnal Rekayasa Elektrika. All right reserved

I. PENDAHULUAN

Wahana udara berpenggerak motor dc *brushless* merupakan hasil rekayasa teknologi yang berkembang dengan pesat. Kebutuhan akan wahana dalam bidang ekonomi, pertahanan, sosial maupun *entertainment* dapat dilengkapi dengan bantuan wahana udara. Sepasang motor berotasi searah jarum jam dan sepasang motor lainnya berotasi berlawanan arah dengan jarum jam. Wahana udara memiliki pengendalian meliputi kecepatan motor dan sikap wahana [2]. Sikap wahana meliputi gerakan *roll*, *pitch*, *yaw*, dan akselerasi. Kecepatan putaran motor menentukan kecepatan pergerakan wahana udara. Pada

wahana udara dilengkapi dengan sensor inersia untuk menjaga kesetimbangan posisi pada saat bergerak yang terintegrasi pada kontroler. Pengendalian menggunakan *remote control* berbasis sinyal infra merah maupun *bluetooth* merupakan metode pengendalian nirkabel dengan jarak pendek. Pemanfaatan sensor sebagai *remote* pengendali wahana dapat dikembangkan untuk wahana nirawak. Dengan memanfaatkan teknologi sensor IMU sebagai sumber masukan memungkinkan diketahui posisi dan percepatan gerakan tangan [3]. *Remote acceleration* (*remote gesture inertia*) menjadi bentuk pengembangan pengendalian *radio control* wahana menjadi lebih sederhana. Bagian utama pengendali adalah sensor

akselerometer yang dilengkapi dengan mikrokontroler. Hasil pembacaan sensor akselerometer dijadikan input *quadcopter* yang merupakan representasi gerakan tangan operator *gesture recognition* [5]. Selain digunakan untuk menentukan kecepatan pergerakan *quadcopter*, data dari *remote gesture* perlu algoritma *template matching* [7] dan *trayektori estimation* [1] serta proses *filter* untuk menghasilkan sebuah gerakan yang sesuai dengan perintah gerakan tangan.

II. STUDI PUSTAKA

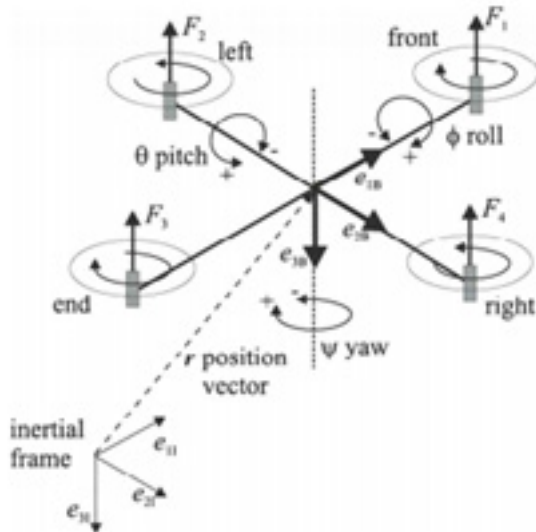
A. Wahana Udara

Quadcopter adalah wahana terbang tanpa awak dengan kemampuan *take-off* dan *landing* secara vertikal (VTOL). *Quadcopter* memiliki 2 *frame* yang beroperasi, diantaranya *inertial frame* yang didefinisikan sebagai *ground* atau area terbang dimana gravitasi ada pada sumbu z negatif dan *body frame* dan titik aksis dari rotor mengarah pada sumbu z positif dan lengan dari *quadcopter* pada sumbu x dan y. Gambar 1 memperlihatkan *frame* pada *quadcopter*.

Posisi dan kecepatan dari *quadcopter* pada *inertial frame* dapat didefinisikan sebagai $x=(x,y,z)^T$ dan $\dot{x}=(\dot{x},\dot{y},\dot{z})^T$. Sedangkan *roll*, *pitch*, dan *yaw* dari *body frame quadcopter* dapat didefinisikan sebagai berikut $\theta=(\phi,\theta,\psi)^T$ dengan catatan $\omega \neq \dot{\theta}$. *Angular velocity* merupakan vektor pada sumbu putar, dimana adalah turunan waktu dari *yaw*, *pitch*, dan *roll*.

$$\omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}, \quad (1)$$

dimana ω adalah *angular velocity* pada *body frame quadcopter*. Relasi dari *body* dan *inertial frame* dapat dilakukan dengan melakukan rotasi matriks R dari *body frame* menuju *inertial frame*. Matriks ini menggunakan



Gambar 1. Frame pada *quadcopter* []

XYZ konvensi *euler angle*:

$$R = \begin{bmatrix} c\phi c\psi - c\theta s\phi s\psi & -c\psi s\phi - c\phi c\theta s\psi & s\theta s\psi \\ c\theta c\psi s\phi + c\phi s\psi & c\phi c\theta c\psi - s\phi s\psi & -c\psi s\theta \\ s\phi s\theta & c\phi s\theta & c\theta \end{bmatrix}, \quad (2)$$

dimana vektor \bar{v} pada *body frame* dan $R\bar{v}$ pada *inertial frame*. Rotasi matriks R adalah *orthogonal*, sehingga $R^{-1} = R^T$ yang merupakan matriks rotasi dari *inertial frame* ke *body frame*. *Quadcopter* yang dibuat, diasumsikan memiliki struktur simetris dengan empat lengan sejajar dengan sumbu x dan sumbu y. Dengan demikian, matriks inersia adalah matriks diagonal I dimana $I_{xx} = I_{yy}$.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (3)$$

B. Hand Motion Remote

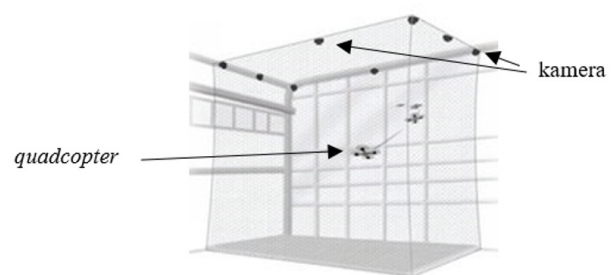
Quadcopter yang dikendalikan menggunakan tangan telah dikembangkan menggunakan beberapa kamera untuk mendeteksi gerakan tangan dan pengendalian gerakan pada *quadcopter*[3]. Pada pengembangan tersebut *quadcopter* mampu bergerak lebih bebas dan fleksibel dibandingkan dengan *remote RC*. Pada pengembangan ini mempunyai keterbatasan yakni *quadcopter* mempunyai area gerak yang terbatas atau bergerak di dalam ruangan yang telah dilengkapi kamera. Gambar 2 Berikut ialah ilustrasi area terbang *quadcopter* dengan *hand motion remote*.

III. METODE

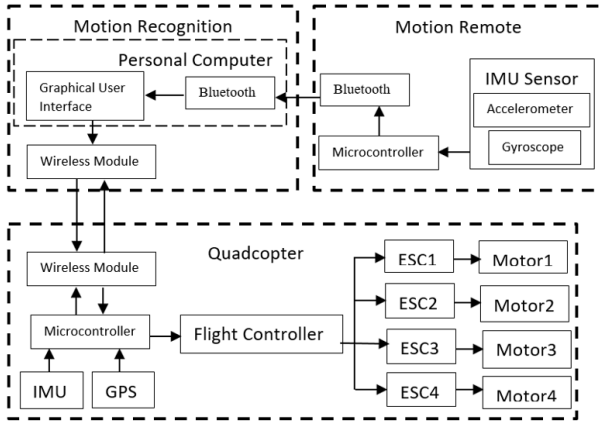
A. Sistem Pengendali Gestur Inersia

Penelitian ini bertujuan merekognisi terhadap gerakan yang dilakukan oleh pengendali (*motion remote*) untuk mengendalikan *quadcopter*. Pada penelitian ini terdapat 3 sub bagian yaitu *quadcopter* sebagai wahana udara, perekognisi gerakan pada *ground station* dan *motion remote* dengan sensor IMU (*Inertial Measurement Unit*) beserta mikrokontroler sebagai pemroses data. Adapun bentuk diagram kerja sistem pengendalian *quadcopter* ditunjukkan pada Gambar 3.

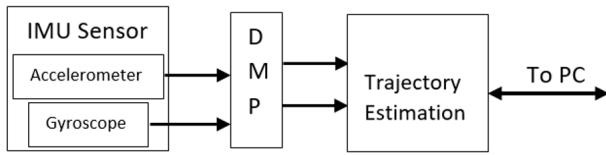
Pada sistem rekognisi terdiri dari dua buah algoritma yaitu estimasi trayektori dan rekognisi gestur. Algoritma



Gambar 2. Area terbang *hand motion remote control*



Gambar 3. Blok diagram rekognisi motion gerakan kontrol quadcopter



Gambar 4. Blok diagram program pada remote

estimasi trayektori menggunakan data akselerasi dan *attitude* dari sensor IMU (akselerometer dan giroskop) kemudian mengubah menjadi trayektori gerakan tangan. Algoritma rekognisi gestur akan mengklasifikasikan trayektori gerakan menjadi sebuah gestur yang telah didefinisikan.

B. Algoritma Estimasi Trayektori

Algoritma estimasi trayektori mengkonversi raw data sensor menjadi data trayektori dalam ruang 3-D serta plot pada bidang 2-D sesuai dengan bidang rekognisi yang ditentukan. Pada Gambar 5 ditunjukkan sistem koordinat pengendali, dimana diterapkan teori INS yaitu Koordinat Bodi (**b**) dan Koordinat Navigasi (**n**) dalam ruang 3-D. Koordinat navigasi (**n**) ialah titik *start* awal dari gerakan *trayektori* yang dilakukan. Sumbu X_n , Y_n , dan Z_n saling tegak lurus satu dengan yang lain dimana sumbu arah Z_n sejajar dengan arah gravitasi bumi. Koordinat Bodi (**b**) ialah koordinat dari bodi *remote* sehingga nilainya tidak akan berubah oleh adanya gerakan. Dan koordinat (**b**) akan sejajar dengan koordinat dari sensor IMU yang digunakan. Sumbu X_b , Y_b , dan Z_b saling tegak lurus satu dengan yang lain dimana sumbu arah Z_b sejajar dengan fisik bodi tongkat pengendali.

Gravitasi pada tiap aksis koordinat body didapatkan dengan memanfaatkan fitur digital *motion processing* dengan persamaan sebagai berikut:

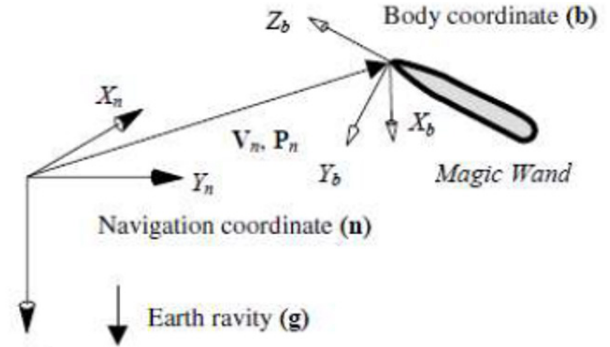
$$\text{Quaternion from DMP} : q = (q_w, q_x, q_y, q_z)^T$$

$$G_x = 2(q_x q_z - q_w q_y), \quad (4)$$

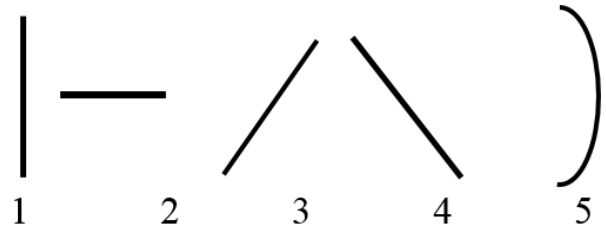
$$G_y = 2(q_w q_x + q_y q_z), \quad (5)$$

$$G_z = q_w q_w - q_x q_x - q_y q_y + q_z q_z, \quad (6)$$

dimana : G_x , G_y , G_z ialah gravitasi tiap aksis.



Gambar 5. Sistem koordinat tongkat pengendali []



Gambar 6. Gerakan dasar 2D

Untuk memperoleh data real akselerasi juga diambil data akselerasi yang terbaca pada sensor akselerometer yang disimbolkan sebagai,

$$Areal_x = A_x - G_x, \quad (7)$$

$$Areal_y = A_y - G_y, \quad (8)$$

$$Areal_z = A_z - G_z, \quad (9)$$

dimana $Areal_x$, $Areal_y$, $Areal_z$ ialah nilai akselerasi tanpa terpengaruh gravitasi.

Data input berasal dari hasil transfer mikrokontroler dan diolah pada *ground station* yang terdiri dari data akselerasi, kecepatan, posisi, dan orientasi. Bentuk gerakan rekognisi yang dimaksudkan ditunjukkan pada Gambar 6. Gerak dasar rekognisi dikelompokkan mejadi lima gerakan, yaitu :

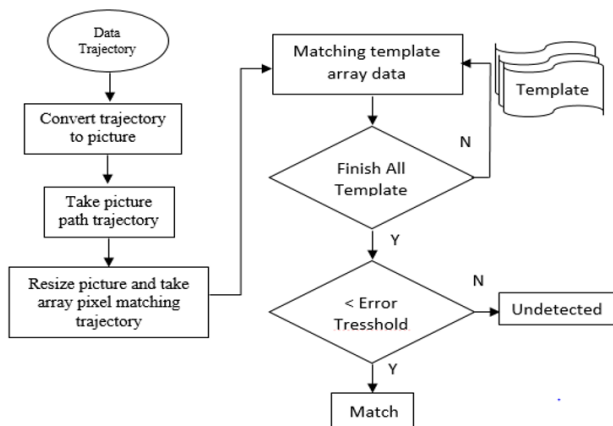
1. Gerakan lurus horizontal
2. Gerakan lurus vertikal
3. Gerakan diagonal arah kanan
4. Gerakan diagonal arah kiri
5. Gerakan parabola/melengkung

Gerakan yang telah direkognisi akan ditampilkan pada *graphical user interface* (GUI) berupa grafik gerakan dasar.

C. Algoritma Template Matching

Sistem rekognisi gestur inersia dirancang menggunakan algoritma *picture based Template Matching* untuk mengenali sebuah gerakan dengan algoritma ditunjukkan pada Gambar 7.

Data *trayektori* yang dikirimkan oleh mikrokontroler akan diubah kedalam domain gambar yang kemudian

Gambar 7. Diagram algoritma *template matching*

diproses hingga mendapat gambar dengan ukuran 10x10 *pixel* biner. Gambar kemudian akan dibandingkan dengan semua *template* yang juga berukuran 10x10, apabila ditemukan hasil *match error* dibawah *treshold* yang ditentukan maka gambar dikenali sebagai gambar *template*. Gambar 8 merupakan gambar hasil deteksi gerakan dimana bagian tengah *crop* terfokus dan bernilai 10x 10 piksel.

D. Wahana udara dan komunikasi

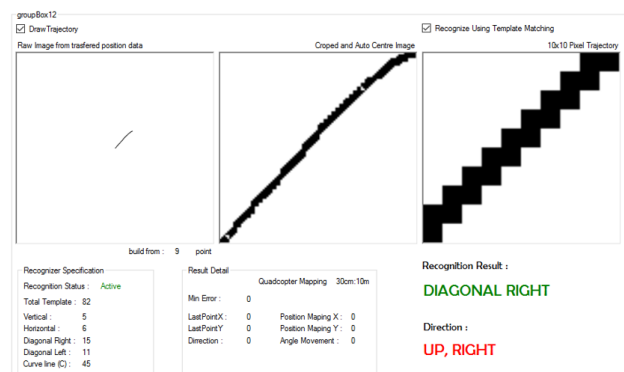
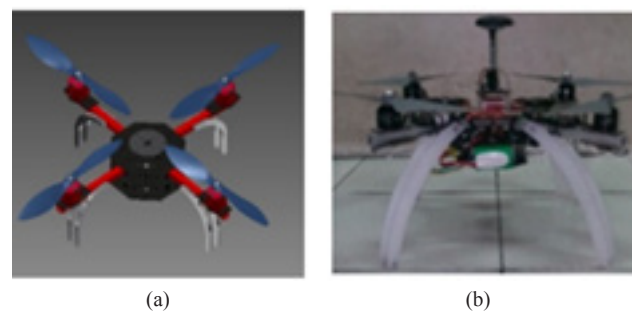
Desain *Quadcopter* yang digunakan pada penelitian ini berukuran 27cm x 27cm dan tinggi 20cm dengan model rangka yang berbentuk menyilang (*X frame*) yang ditunjukkan pada Gambar 9 sedangkan data header untuk perintah gerak disajikan pada Tabel 1.

IV. HASIL DAN PEMBAHASAN

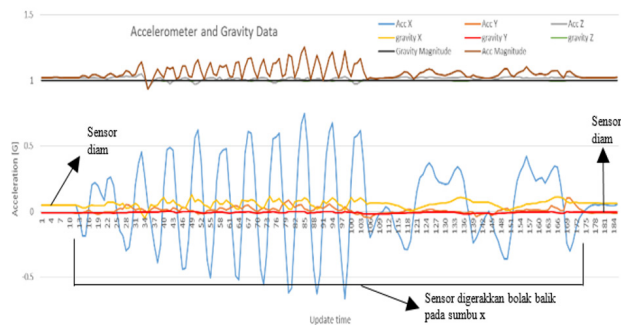
Data-data sensor perlu diperoleh terlebih dahulu untuk keperluan kalibrasi. Data sensor IMU yaitu sensor akselerometer merupakan data utama yang diperlukan sebagai referensi.

Dari Tabel 2 ditunjukkan saat keadaan diam nilai data sensor akselerometer bernilai sama dengan gravitasi sehingga riil akselerasi bernilai mendekati 0. Pada saat mengukur sumbu X data riil akselerasi tidak sama dengan nol sehingga diperlukan *threshold* untuk menentukan sensor dalam keadaan diam atau sedang bergerak.

Nilai gravitasi pada setiap aksis relatif konstan meskipun sensor digerakkan pada sumbu X secara bolak-balik dengan cepat (Gambar 10). *Magnitude* dari gravitasi terlihat konstan sebesar 1G tidak seperti *magnitude accelerometer* yang berubah saat ada gerakan. Akselerasi pada sumbu X terbaca bahwa terdapat gerakan bolak-balik namun selama orientasi tidak berubah maka pembacaan gravitasi relatif konstan, pada gambar terukur ada perubahan kecil nilai gravitasi saat sensor bergerak. Hal ini disebabkan algoritma *sensor fusion* yang mendeteksi ada perubahan orientasi karena adanya lonjakan data dari akselerometer namun algoritma yang dipakai dapat meredam efek perubahan akselerasi dengan cukup baik.

Gambar 8. Hasil deteksi gerakan kiri: raw data; tengah *crop* terfokus; 10x10 *pixel*Gambar 9. (a) Rancangan (b) Realisasi *quadcopter*Tabel 1. Rancangan data header yang dikirimkan ke *quadcopter*

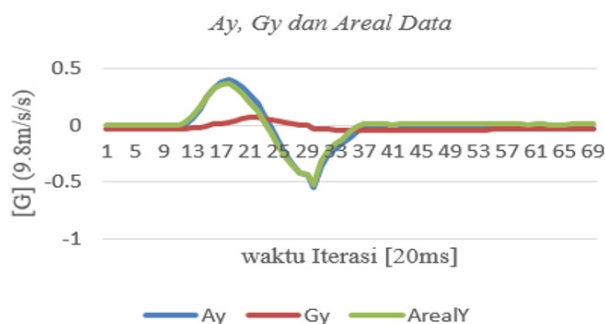
Gerakan	Arah	Header	Respon/gerakan yang diharapkan
Horizontal	→	'K'	Quadcopter bergerak ke kanan
	←	'L'	Quadcopter bergerak ke kiri
Vertikal	↑	'M'	Quadcopter bergerak ke belakang
	↓	'N'	Quadcopter bergerak ke depan
Diagonal kanan	↗	'V'	Quadcopter bergerak ke kanan belakang
	↘	'W'	Quadcopter bergerak ke kiri depan
Diagonal Kiri	↖	'X'	Quadcopter bergerak ke kiri belakang
	↙	'Y'	Quadcopter bergerak ke kanan depan
Curve/ melengkung (C)	↻	'Z'	Quadcopter bergerak melengkung ke depan
Gerakan	Arah	Header	Respon/gerakan yang diharapkan
Command	-	'ID'	Quadcopter mengirimkan data String "Iquadcopter"
Command	-	'A'	Quadcopter Landing
Command	-	'B'	Putar/stop motor
Command	-	'C'	Take off
Command	-	'D'	Hover



Gambar 10. Nilai gravitasi bumi pada tiap aksis

Tabel 2. Tabel hubungan A, G, dan Areal

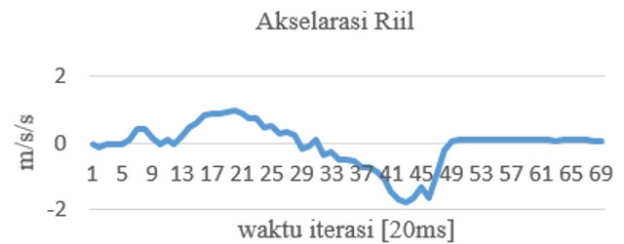
Sumbu pengukuran	Sudut pengukuran	Data akselerometer	Data gravitasi	Data real akselerasi
Y	± 0	-0.04	-0.04	0
Y	± 30	0.49	0.49	0
Y	± 60	0.87	0.87	0
Y	± 90	1	1	0
X	± 0	0.01	-0.01	0.02
X	± 30	0.53	0.52	0.02
X	± 60	0.93	0.89	0.04
X	± 90	1.04	1	0.04



Gambar 11. Hubungan data sensor akselerasi

Pada Gambar 11 ditunjukkan gambar grafik hubungan antara A_y , G_y dan $Areal_y$ saat sensor dalam keadaan bergerak ke arah Y positif. Pengujian sensor digerakkan ke arah sumbu Y positif dengan kondisi sensor sejajar bumi (± 00). Pada grafik saat iterasi 0-10, sensor dalam keadaan tidak bergerak, semua komponen pengukuran bernilai mendekati nol sedangkan dari iterasi 11-37 sensor digerakkan ke arah sumbu Y positif terukur bahwa nilai dan bernilai positif (mengalami percepatan) dan kemudian berubah menjadi negatif (mengalami perlambatan) dan kembali diam, nilai G_y relatif konstan dikarenakan G_y ialah nilai gravitasi dan G_y nilai akan bernilai konstan selama orientasi sensor tidak berubah jika mungkin.

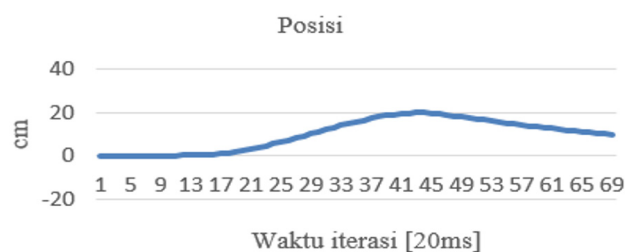
Data riil akselerasi yaitu data akselerasi sesungguhnya tanpa terpengaruh percepatan gravitasi, data real akselerasi didapat dari pengolahan data attitude dan data akselerasi yang dibaca oleh sensor (algoritma estimasi *trayektori*). Pengujian ini dilakukan dengan menggerakkan sensor ke arah sumbu Y positif sensor, grafik Gambar 12 ialah



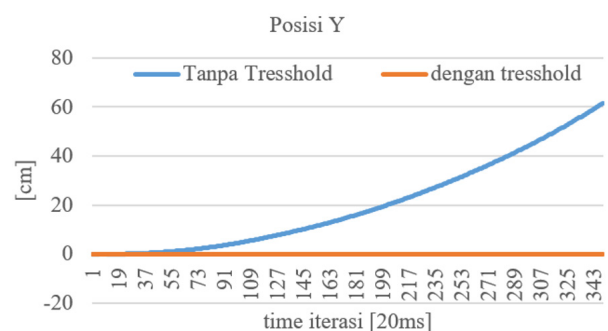
Gambar 12. Grafik riil akselerasi



Gambar 13. Grafik kecepatan (integral dari riil akselerasi)



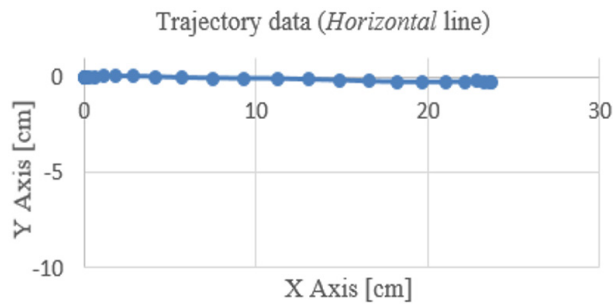
Gambar 14. Grafik posisi (integral dari kecepatan)

Gambar 15. Grafik posisi dengan dan tanpa *threshold*

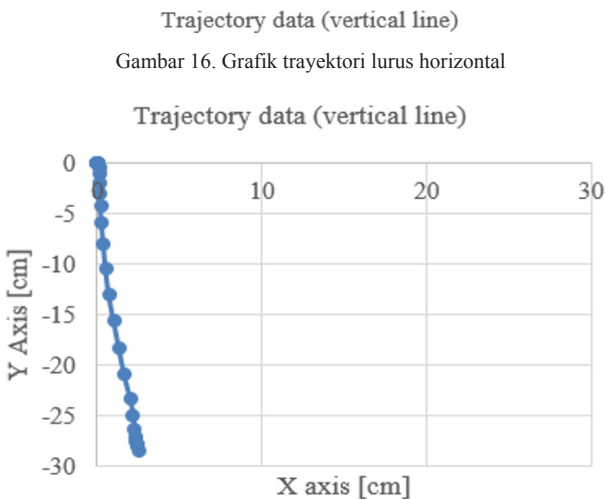
hasil pembacaan data *real* akselerasi. Pada Gambar 13 ditunjukkan grafik kecepatan yang didapat dari pengintegralan nilai *real* akselerasi yang bersumber dari data tersaji pada Gambar 12. Dan Gambar 14 ialah gambar grafik perpindahan posisi yang didapat dari hasil pengintegralan ganda data *real* akselerasi.

Tiga grafik (Gambar 12, Gambar 13, dan Gambar 14) menunjukkan hubungan antara *real* akselerasi, kecepatan, dan posisi. Pada grafik posisi iterasi ke 43 terjadi posisi diam wahana udara namun pada aktual pengukuran terjadi pergeseran ke arah negatif (posisi berkurang). Hal ini disebabkan oleh *error* pengintegralan ganda (Gambar 15 iterasi 43) dan akan terus mengakumulasi *error* integral.

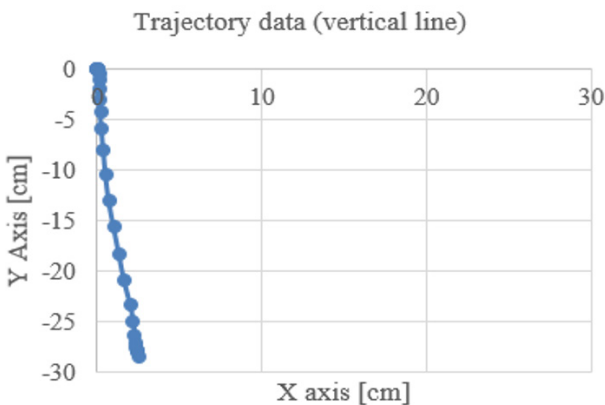
Efek *error* integral tidak begitu terlihat dikarenakan estimasi posisi hanya dicari saat adanya penekanan tombol (*pushbutton*) pada *remote gesture* inersia atau *remote* kendali dan waktu penggerakan sensor juga relatif



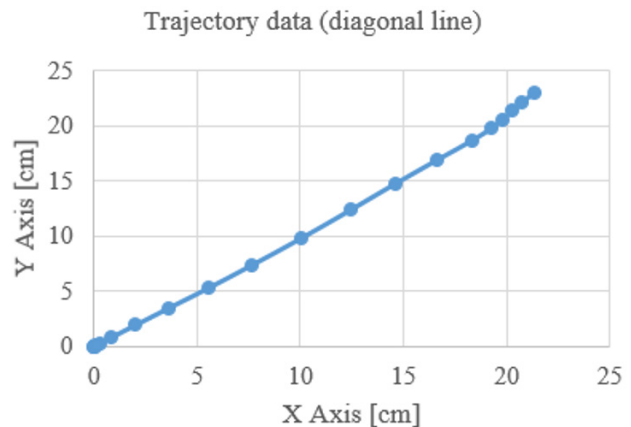
Gambar 16 Grafik trayektori lurus horizontal



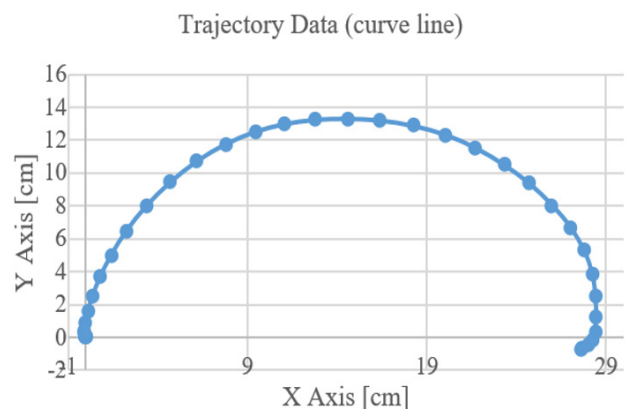
Gambar 16. Grafik trayektori lurus horizontal



Gambar 17. Data trayektori garis lurus vertikal



Gambar 18. Data trayektori garis lurus diagonal



Gambar 19. Data trayektori garis melengkung terhadap sumbu x

pendek, selain itu pada penelitian ini juga digunakan *threshold* gerakan untuk mengkompensasi *error* integral yang hasilnya ditunjukkan pada Gambar 15.

Karakterisasi pengendali gerakan wahana udara memanfaatkan sensor akselerometer yang berada pada sensor *Inertial Measurement Unit*. Untuk mendapatkan gerakan lurus digunakan media pengujian yang menjaga gerakan tangan lurus dengan kecepatan yang berbeda.

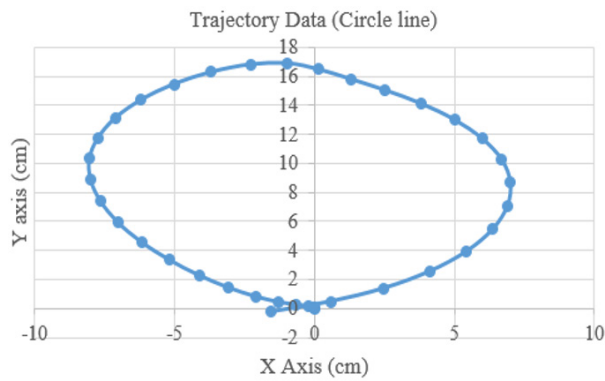
Data pada Gambar 16 terjadi *error drift* sumbu Y yang kecil yakni sebesar 0,25 cm dan sudah dapat dihilangkan. *Error drift* ini dapat disebabkan masih terdapat *noise* pembacaan real akselerasi pada sumbu Y yang perlu dihilangkan. Selain itu alat bantu ukur yang dipakai juga tidak dapat mengatur atau menentukan arah hadap sensor dengan pasti sehingga memungkinkan arah hadap sensor yang sedikit miring atau tidak sepenuhnya horizontal terhadap lintasan lurus yang digunakan untuk pengambilan data sehingga terdapat pembacaan real akselerasi pada sumbu Y, sedangkan pada pembacaan data pada sumbu X, panjang garis horizontal yang terdeteksi yakni sekitar 24 cm yang berarti terdapat *error* sebesar 6 cm terhadap referensi (30cm) hal ini juga mungkin disebabkan pengolahan data untuk mendapatkan nilai real akselerasi yang kurang baik serta arah hadap sensor belum sepenuhnya tepat, selain itu proses pengintegralan ganda memang dikenal menghasilkan *error* yang semakin besar seiring berjalannya waktu.

Pengambilan data secara vertikal ditunjukkan pada

Gambar 17 sama halnya dengan pengambilan data horizontal digerakkan sepanjang 30cm, namun perbedaan terdapat pada arah hadap sensor yang diubah menjadi vertikal. Terjadi kesalahan *error drift* sumbu x sekitar 2.5 cm, hal ini dikarenakan terdapat *noise* pada pembacaan akselerasi riil. Arah hadap sensor tidak tegak lurus vertikal menjadi faktor utama terjadinya *error drift* pada saat pembacaan akselerasi riil sumbu X, sedangkan pembacaan pada sumbu Y, panjang garis horizontal terukur 28 cm dengan kesalahan terukur 2cm terhadap referensi 30cm.

Grafik pada Gambar 18 menunjukkan titik akhir dari *trayektori* diagonal berada pada (21.3, 23) dari titik tersebut maka dapat dihitung panjang garis diagonal yang terlihat pada gambar dengan rumus pitagoras $x^2+y^2=k^2$ dengan x = letak titik akhir sumbu X, y = letak titik akhir sumbu Y, dan k ialah panjang garis diagonal yang dicari. Sehingga $21.3^2+23^2=k^2$; $k = 31.3\text{cm}$ yang berarti *error* jarak yang terjadi terhadap referensi (30cm) sebesar 1,3cm. Sama seperti dua data sebelumnya hal ini juga mungkin disebabkan pengolahan data untuk mendapatkan nilai *real* akselerasi yang kurang baik, selain itu proses pengintegralan ganda memang dikenal menghasilkan *error* yang semakin besar seiring berjalannya waktu.

Pada Gambar 20 (trayektori lingkaran) titik awal garis yakni berada pada titik (0,0) dan kemudian membentuk gerakan memutar searah jarum jam dan berakhir pada titik (-1,5, 0). Tidak seperti percobaan horizontal, vertikal dan diagonal yang memiliki referensi yang pasti, gerakan



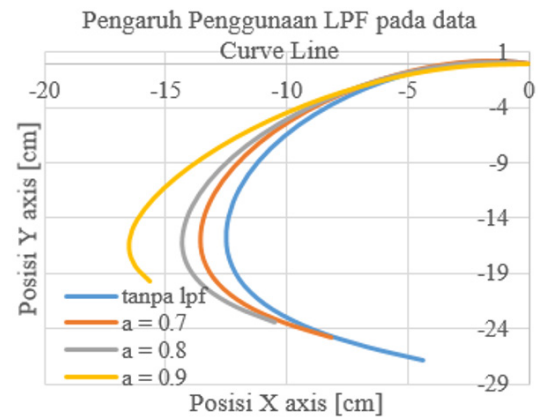
Gambar 20. Data trayektori melingkar

Tabel 3. Perbandingan penggunaan *low pass filter* pada estimasi trayektori

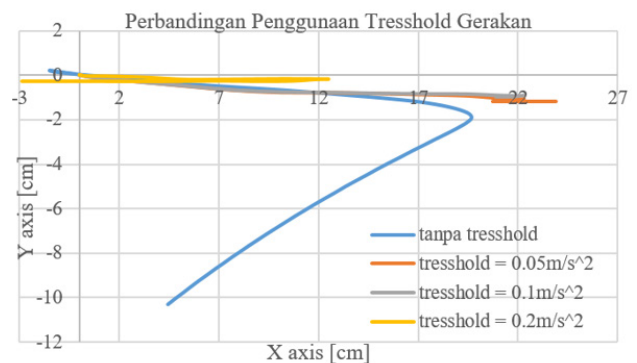
iterasi	LPF $\alpha = 0.3$		Tanpa LPF	
	x [cm]	y [cm]	x [cm]	y [cm]
1	0	0	0	0
2	0	0	0	0
3	0.01	0	0.02	0
4	0.06	0.01	0.10	0.02
5	0.18	0.03	0.30	0.05
6	0.42	0.06	0.67	0.10
7	0.84	0.12	1.24	0.17
8	1.48	0.19	2.08	0.26
9	2.36	0.29	3.15	0.39
10	3.40	0.43	4.34	0.55
11	4.56	0.59	5.61	0.72
12	5.82	0.76	6.99	0.91
13	7.24	0.94	8.61	1.08
14	8.91	1.12	10.52	1.27
15	10.78	1.31	12.55	1.48
16	12.69	1.53	14.47	1.72
17	14.54	1.77	16.31	1.96
18	16.30	2.01	18.01	2.19
19	17.92	2.24	19.54	2.41
20	19.39	2.47	20.94	2.64
21	20.70	2.71	22.11	2.89
22	21.78	2.97	23.01	3.15
23	22.66	3.23	23.78	3.41
24	23.43	3.50	24.49	3.67
25	24.13	3.77	25.17	3.95
26	24.80	4.05	25.83	4.24

pada percobaan ini hanya menggunakan referensi gerakan melingkar yang dilakukan oleh tangan manusia, sehingga tidak dapat diketahui dengan pasti *error* aktual yang terjadi, akan tetapi dari data Gambar 20 diatas terlihat bahwa lintasan berbentuk lingkaran sesuai dengan referensi yang diberikan.

Pengujian *filter* untuk trayektori gerakan menggunakan *low pass filter* pada data output sensor akselerometer menggunakan fitur DMP dibandingkan dengan tanpa



Gambar 21. Pengaruh *low pass filter* pada data gerakan melengkung



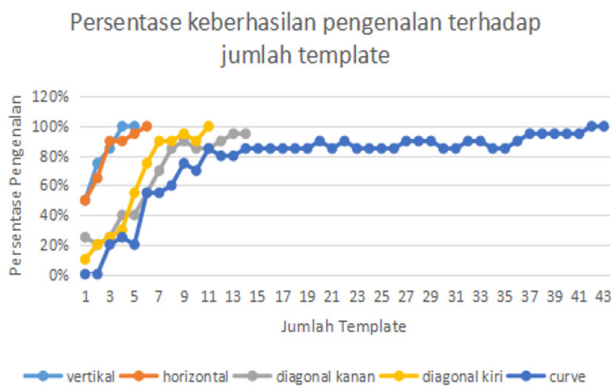
Gambar 22. Pengaruh penggunaan *threshold* gerakan pada gerak horizontal

adanya *filter*. Data keluaran dari fitur DMP (tanpa *filter*) mempunyai trayektori yang lebih akurat dalam hal panjang searah dengan referensi (gerakan ke arah sumbu X sejauh 30cm) dengan perbedaan 1,03cm lebih panjang dari data terfilter, akan tetapi terjadi pergeseran 0,19cm lebih besar dibandingkan dengan data dengan menggunakan *low pass filter* dengan nilai $\alpha = 0,3$ ditunjukkan pada Tabel 3.

Perbandingan penggunaan *low pass filter* dengan nilai α yang berbeda-beda pada gerakan *curve*/melengkung. Pada Gambar 21 ditunjukkan bahwa dari seluruh pengujian penggunaan *low pass filter* ini diketahui bahwa semakin kecil nilai α maka trayektori yang dibangun akan semakin mendekati data asli (tanpa *low pass filter*) dikarenakan sinyal keluaran *low pass filter* akan mendekati *input* dengan semakin kecilnya nilai α .

Data pada Gambar 21 menunjukkan bahwa penggunaan *filter* tidak selalu menghasilkan data yang lebih baik, hal ini perlu disesuaikan dengan kebutuhan sistem yang dibangun. Penggunaan *threshold* mempunyai kelebihan terutama dalam mengurangi efek akumulasi eror integral saat dalam keadaan diam namun juga mempunyai kekurangan yakni sistem menjadi tidak dapat membaca gerakan-gerakan dengan nilai yang kurang dari *threshold* (gerakan pelan) seperti yang terjadi pada Gambar 22 garis kuning (*threshold* sebesar 0,2m/s atau 20cm/s) gerakan yang dibaca menjadi sangat pendek dan jauh dari referensi yang diberikan yakni gerakan sepanjang 25cm.

Pada data tanpa *threshold* tersaji pada Gambar 22



Gambar 23. Persentase keberhasilan pengenalan terhadap jumlah template (garis lurus vertikal)

Tabel 4. Data persentase keberhasilan pengenalan garis (garis curve C)

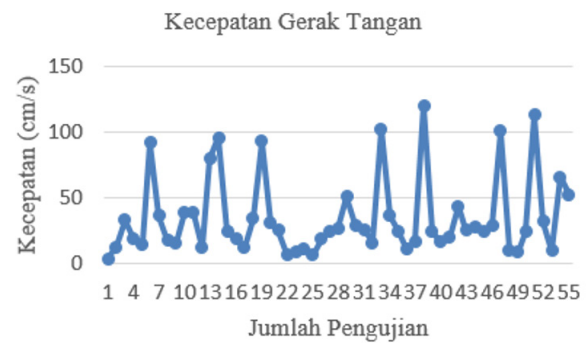
Gerakan	Hori- zon	Verti- kal	Diag- onal kanan	Diag- onal kiri	Curve	Total
Template	6	5	14	11	43	79
Jumlah gerakan	30	30	30	30	30	150
Jumlah salah	0	0	2	0	4	6
Jumlah Benar	30	30	28	30	26	144
Persentase benar	100%	100%	93%	100%	87%	96%

(garis biru) tidak dapat dikategorikan ke dalam garis horizontal dikarenakan terjadi pembelokan menuju y aksis negatif dan nilai x juga semakin menuju negatif yang berarti kecepatan gerak pada akhir garis tanpa *threshold* juga bernilai negatif dan semakin lama semakin besar karena terus mengakumulasi kesalahan (*error integral ganda*). Pada garis abu-abu dan oranye lintasan/trayektori yang dihasilkan mempunyai panjang dan bentuk yang mendekati/sesuai dengan referensi. Hal ini dikarenakan nilai *threshold* yang baik (tidak terlalu besar/kecil) sehingga mampu untuk meredam efek *error integral* dan mampu membaca gerakan yang relatif pelan.

A. Pengujian Rekognisi Gerakan

Pengujian pengambilan pengenalan gerakan ini dibagi menjadi enam sub pengujian yakni pengujian pengenalan garis vertikal, pengenalan garis horizontal, pengenalan garis diagonal kanan, pengenalan garis diagonal kiri, pengenalan garis melengkung/parabola/*curve* dan pengujian keseluruhan proses pengenalan.

Data didapat dari menggerakkan *remote/sensor* berupa gerakan garis vertikal (pada bidang X dan Y) yang dilakukan dengan cara menggerakkan *remote* dengan menggunakan tangan manusia (*free hand movement*) yang kemudian direkognisi menggunakan algoritma *template matching*. Semakin banyak jumlah *template* maka persentase ketepatan juga semakin meningkat. Namun demikian di titik/jumlah tertentu ketepatan sudah tidak bertambah/konstan (maksimum) sehingga apabila



Gambar 24. Pengujian *remote* pengendali *gesture inertia*

template ditambah lagi dengan pola vertikal yang lain maka tidak akan berpengaruh pada presentase ketepatan namun justru akan menambah proses/komputasi pengenalan yang berlebih sesuai dengan proses pengenalan.

Untuk setiap jumlah *template* dilakukan gerakan sebanyak 20 kali kemudian dihitung presentasi kecocokan terhadap referensi gerakannya (gerakan vertikal) yang ditunjukkan pada Gambar 23. Pengujian *template matching* horizontal memiliki hasil yang tidak jauh berbeda dengan pengujian pada *template* vertikal. Data pengujian *template* horizontal disajikan pada Gambar 23.

Pengujian garis diagonal kanan ini didapati bahwa jumlah *template* yang digunakan lebih banyak dibandingkan garis vertikal maupun horizontal, hal ini dikarenakan variasi *pixel* yang dapat menyusun sebuah garis diagonal lebih banyak dibandingkan dengan garis horizontal atau garis vertikal. Pada pengujian ini juga didapat bahwa seiring bertambahnya jumlah *template* yang digunakan persentasi rekognisi juga semakin tinggi. Pada saat *template* yang digunakan berjumlah delapan, data mulai menunjukkan keadaan yang konstan yakni ketepatan >85%, pada pengujian dengan jumlah *template* sebesar 14 persentase pengenalan yang didapat yakni sebesar 95% yang ditunjukkan pada Gambar 23. Pengujian untuk *template matching* diagonal ke kiri memiliki nilai prosentase yang tidak jauh berbeda dengan pengenalan *template matching* pada gerakan diagonal ke kiri. Namun kesamaan sudah diperoleh pada saat jumlah *template* 11 dan ditunjukkan pada Gambar 23.

Pengujian pengenalan garis *curve* menggunakan gerakan yang berbeda dengan empat macam gerakan lurus yang sebelumnya.

Pada Tabel 4 ditunjukkan hasil pengujian yang dilakukan dengan jumlah *template* tertinggi yang digunakan yakni sebanyak 43 *template* dan menghasilkan persentasi ketepatan sebesar >95%. Pada Gambar 23 ditunjukkan data grafik persentasi ketepatan pengenalan terhadap jumlah *template* untuk gerakan melengkung/*curve*.

Pengujian kecepatan *remote gesture inertia* untuk kecepatan gerakan quadcopter tersaji pada Tabel 5. Pengujian gerakan *remote gesture inertia* yang ditunjukkan pada Gambar 24 sebagai pengendali dimaksudkan agar diperoleh data referensi untuk konversi ke dalam bentuk kecepatan wahana udara. Pengujian dilakukan sebanyak

Tabel 5. Pengujian remote gesture inertia untuk kecepatan *quadcopter*

Arah Gerak	Gerakan Tangan		Quadcopter	
	Jarak (cm)	Kecepatan (cm/s)	PWM (%)	Kecepatan (cm/s)
Kiri	23.2	18.142	68.33	92.5
	29.41	71.886	179.73	243.3
	30.23	104.784	285.64	520.6
	36.87	122.023	375.16	507.2
	58.13	127.283	419.80	566.8
Kanan	12.67	30.056	92.24	124.47
	47.78	26.921	105.57	142.83
	39.98	76.04	252.58	340.9
	122.1	30.652	302.79	461.9
Depan	60.23	127.627	420.45	630.6
	32.45	35.128	120.17	162.30
	20.56	97.395	201.46	260.9
	41.77	80.62	263.58	350.8
	60.12	42.822	300.66	405.88
Belakang	37.98	124.833	393.68	531.7
	40.12	27.756	98.62	132.53
	24.22	98.368	234.39	316.58
	42.43	77.728	260.39	351.76
	61.65	44.087	353.65	477.58
Diagonal	48.77	145.438	438.12	580.6
	20.17	15.904	64.33	86.58
	28.14	40.471	140.36	189.4
	40.65	70.952	244.39	330.1
	62.88	42.294	315.33	425.2
	70.18	70.9	430.06	582.5

lima puluh lima kali dengan gerakan pendek agar tidak ada *drifting* data. Hasil pengujian menunjukkan 3,14 cm/s kecepatan tangan terendah dan 120,5 cm/s kecepatan tangan tertinggi yang mampu dibaca oleh sensor dan rata-rata gerakan di 37,8 cm/s. Konsistensi kecepatan gerakan dan kualitas sensor menentukan output gerakan kecepatan wahana udara oleh motor penggerak. Ketika *input* jarak pendek (23,2 cm) dan kecepatan tangan lambat (18,12cm/s) maka *output* berupa pwm adalah pwm kecil (68,3). Data yang dihasilkan di atas sudah sesuai dengan aturan *fuzzy* yang telah dibuat sebelumnya.

V. KESIMPULAN

Pada penelitian ini telah dilakukan pengujian trayektori *quadcopter* dimana lintasan yang ditempuh menggunakan algoritma estimasi trayektori cukup akurat dengan hasil *output* gerakan yang menyerupai referensi pada gerak lurus berjarak 30cm. Gerakan tangan sebagai *remote* dapat digunakan untuk mengendalikan *quadcopter* khususnya kecepatan gerak sesuai dengan kecepatan gerakan tangan. Pengenalan gerakan menggunakan algoritma *template matching* memiliki persentase keberhasilan 96% dimana jumlah *template* mempengaruhi persentase keberhasilan.

REFERENSI

- [1] Beh, J., Han, D. & Ko, H. *Rule based trajectory segmentation for modelling hand motion trajectory*. Pattern Recognition, 47(4), 1586-1601. (2014).
- [2] Tamami, Niam, Endra Pitowarno, and I. Gede Puja Astawa. *Proportional Derivative Active Force Control for "X" Configuration Quadcopter*. Journal of Mechatronics, Electrical Power, and Vehicular Technology 5, no. 2 (2014): 67-74.
- [3] Cho, S.J., Oh, J.K., Bang, W.C., Chang, W., Choi, E., Jing, Y., Cho, J. and Kim, D.Y. *Magic wand: a hand-drawn gesture input device in 3-D space with inertial sensors*. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on* pp. 106-111. IEEE, (2004), October.
- [4] Xu, R., Zhou, S. and Li, W.J. *MEMS accelerometer based nonspecific-user hand gesture recognition*. *IEEE sensors journal*, 12(5), pp.1166-1173, (2012).
- [5] Cheoljong Yang, David K. Han, Hanseok Ko, *Continuous hand gesture recognition based on trajectory shape information*, pattern recognition letters, 99, 39-47.(2014).
- [6] Andrea, D., Raffaello. "The outstanding athletic power of quadcopter"(2012).
- [7] Roberto, B. *Template matching techniques in computer vision: theory and practice*, Wiley, 2009.
- [8] Hendriawan, Akhmad. "Sistem Kontrol Altitude pada UAV Model Quadcopter dengan Metode PID". *Electronic Engineering Polytechnic Institute of Surabaya*. pp. 271-350, (2012).
- [9] H. Sihombing, Sudirman. "Aplikasi Sensor Accelerometer untuk Menulis di Udara". *Electronical Industry of Diponegoro University*, 2012.
- [10] Do Khac Tiep, Kinam lee, Young-Jae Ryoo, Si Jung kim. *A Fuzzy-PD Controller for Autonomous Aerial Robot*, URAI (2017).

Penerbit:

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala

Jl. Tgk. Syech Abdurrauf No. 7, Banda Aceh 23111

website: <http://jurnal.unsyiah.ac.id/JRE>

email: rekayasa.elektrika@unsyiah.net

Telp/Fax: (0651) 7554336

